

**CLAIMS**

The following is claimed:

- 1           1.       A method for providing encryption for the rerouting of multi-media data flow  
2       packets, comprising the steps of:  
3                assigning a sequence number to a first multi-media data flow packet received by a first  
4       endpoint, wherein said first multi-media data flow packet is within a series of multi-media data  
5       flow packets;  
6                pseudo-randomly shuffling said sequence number of said first multi-media data flow  
7       packet; and  
8                transmitting said pseudo-randomly shuffled sequence number to a second endpoint.
- 1           2.       The method of claim 1, wherein said multi-media data flow packets are real-time  
2       multi-media data flow packets.
- 1           3.       The method of claim 1, wherein said pseudo-random shuffling is performed via  
2       use of randomization code that is algorithmically predictable if a key to said randomization code  
3       is known.
- 1           4.       The method of claim 1, wherein said series of multi-media data flow packets,  
2       including said first multi-media data flow packet, are assigned sequence numbers that are each  
3       pseudo-randomly shuffled prior to said transmitting step.

1           5.       The method of claim 1, further comprising the step of pseudo-randomly shuffling  
2       a destination address of said first multi-media data flow packet.

1           6.       The method of claim 5, wherein said destination address is a destination port  
2       address of said second endpoint.

1           7.       The method of claim 4, further comprising the step of re-sequencing said series of  
2       multi-media data flow packets so that said re-sequenced multi-media data flow packets are  
3       transmitted from said first endpoint to said second endpoint in a random order.

1           8.       The method of claim 7, wherein said re-sequenced multi-media data flow packets  
2       are transmitted within a predefined jitter buffer size.

1           9.       The method of claim 1, further comprising the step of performing bit  
2       manipulation within said first multi-media data flow packet.

1           10.      The method of claim 9, wherein said step of performing bit manipulation is  
2       performed by using a bitsize operation that is restorable.

1           11.      The method of claim 10, wherein said bitsize operation uses a negation operator,  
2       such that every 1 bit becomes a 0 bit and every 0 bit becomes a 1 bit.

1           12.    A system for providing encryption for the rerouting of multi-media data flow  
2 packets, comprising:

3           means for assigning a sequence number to a first multi-media data flow packet received  
4 by a first endpoint, wherein said first multi-media data flow packet is within a series of multi-  
5 media data flow packets;

6           means for pseudo-randomly shuffling said sequence number of said first multi-media  
7 data flow packet; and

8           means for transmitting said pseudo-randomly shuffled sequence number to a second  
9 endpoint.

1           13.    The system of claim 12, wherein said multi-media data flow packets are real-time  
2 multi-media data flow packets.

1           14.    The system of claim 12, wherein said means for pseudo-random shuffling  
2 performs said shuffling via use of randomization code that is algorithmically predictable if a key  
3 to said randomization code is known.

1           15.    The system of claim 12, further comprising means for pseudo-randomly shuffling  
2 a destination address of said first multi-media data flow packet.

1           16.    The system of claim 15, wherein said destination address is a destination port  
2 address of said second endpoint.

1           17.     The system of claim 12, further comprising means for re-sequencing said series of  
2 multi-media data flow packets so that said re-sequenced multi-media data flow packets are  
3 transmitted from said first endpoint to said second endpoint in a random order.

1           18.     The system of claim 17, wherein said re-sequenced multi-media data flow packets  
2 are transmitted within a predefined jitter buffer size.

1           19.     The system of claim 12, further comprising means for performing bit  
2 manipulation within said first multi-media data flow packet.

1           20.     The system of claim 19, wherein said means for performing bit manipulation uses  
2 a bitsize operation that is restorable.

1           21.     The system of claim 20, wherein said bitsize operation uses a negation operator,  
2 such that every 1 bit becomes a 0 bit and every 0 bit becomes a 1 bit.

1           22.    A system for providing encryption for the rerouting of multi-media data flow  
2    packets, comprising:  
3                   a first endpoint, connected to a second endpoint, wherein said first endpoint  
4    comprises;  
5                   a transceiver;  
6                   software stored within said first endpoint defining functions to be performed by  
7    said first endpoint; and  
8                   a processor configured by said software to perform the steps of,  
9                   assigning a sequence number to a first multi-media data flow packet  
10   received by a first endpoint, wherein said first multi-media data flow packet is within a series of  
11   multi-media data flow packets;  
12                   pseudo-randomly shuffling said sequence number of said first multi-media  
13   data flow packet; and  
14                   transmitting said pseudo-randomly shuffled sequence number to a second  
15   endpoint.

1           23.    The system of claim 22, wherein said multi-media data flow packets are real-time  
2    multi-media data flow packets.

1           24.    The system of claim 22, wherein said multi-media data flow packets are real-time  
2    multi-media data flow packets.

1           25.     The system of claim 22, wherein said pseudo-random shuffling is performed via  
2     use of randomization code that is algorithmically predictable if a key to said randomization code  
3     is known.

1           26.     The system of claim 22, wherein said series of multi-media data flow packets,  
2     including said first multi-media data flow packet, are assigned sequence numbers that are each  
3     pseudo-randomly shuffled prior to said transmitting step.

1           27.     The system of claim 22, wherein said processor is further configured by said  
2     software to perform the step of pseudo-randomly shuffling a destination address of said first  
3     multi-media data flow packet.

1           28.     The system of claim 27, wherein said destination address is a destination port  
2     address of said second endpoint.

1           29.     The system of claim 26, wherein said processor is further configured by said  
2     software to perform the step of re-sequencing said series of multi-media data flow packets so that  
3     said re-sequenced multi-media data flow packets are transmitted from said first endpoint to said  
4     second endpoint in a random order.

1           30.     The system of claim 29, wherein said re-sequenced multi-media data flow packets  
2     are transmitted within a predefined jitter buffer size.

1           31.     The system of claim 22, wherein said processor is further configured by said  
2 software to perform the step of performing bit manipulation within said first multi-media data  
3 flow packet.

1           32.     The system of claim 31, wherein said step of performing bit manipulation is  
2 performed by using a bitsize operation that is restorable.

1           33.     The system of claim 32, wherein said bitsize operation uses a negation operator,  
2 such that every 1 bit becomes a 0 bit and every 0 bit becomes a 1 bit.

1           34.     A system for providing encryption for the routing of multi-media data flow  
2 packets, comprising:

3                 a first endpoint connected to a second endpoint, wherein said second endpoint comprises:

4                     a transceiver;

5                     software stored within said second endpoint defining functions to be performed  
6 by said second endpoint; and

7                     a processor configured by said software to perform the steps of:

8                         unshuffling a pseudo-randomly shuffled sequence number received from  
9 said first endpoint, via use of an alogrithmic key; and

10                        deriving a first data flow packet from said unshuffled sequence number,  
11 wherein said first data flow packet is within a series of data flow packets.

1           35.     A system for providing encryption for the routing of data flow packets,  
2     comprising:  
3           a first endpoint connected to a second endpoint, wherein said first endpoint comprises:  
4                 a transceiver; and  
5                 a controller programmed to perform the steps of:  
6                         assigning a sequence number to a first multi-media data flow packet  
7     received by a first endpoint, wherein said first multi-media data flow packet is within a series of  
8     multi-media data flow packets;  
9                         pseudo-randomly shuffling said sequence number of said first data flow  
10    packet; and  
11                         transmitting said pseudo-randomly shuffled sequence number to a second  
12    endpoint.

13  
14           36.     The system of claim 35, wherein said multi-media data flow packets are real-time  
15    multi-media data flow packets.

1           37.     The system of claim 35, wherein said series of multi-media data flow packets,  
2     including said first multi-media data flow packet, are assigned sequence numbers that are each  
3     pseudo-randomly shuffled prior to said transmitting step.



1           38.     The system of claim 35, wherein said controller is further programmed to perform  
2     the step of pseudo-randomly shuffling a destination address of said first multi-media data flow  
3     packet.

1           39.     The system of claim 38, wherein said destination address is a destination port  
2     address of said second endpoint.

1           40.     The system of claim 37, wherein said processor is further configured by said  
2     software to perform the step of re-sequencing said series of multi-media data flow packets so that  
3     said re-sequenced multi-media data flow packets are transmitted from said first endpoint to said  
4     second endpoint in a random order.

1           41.     The system of claim 40, wherein said re-sequenced multi-media data flow packets  
2     are transmitted within a predefined jitter buffer size.

1           42.     The system of claim 35, wherein said controller is further configured to perform  
2     the step of performing bit manipulation within said first multi-media data flow packet.

1           43.     The system of claim 42, wherein said step of performing bit manipulation is  
2     performed by using a bitsize operation that is restorable.

1           44.     The system of claim 43, wherein said bitsize operation uses a negation operator,  
2     such that every 1 bit becomes a 0 bit and every 0 bit becomes a 1 bit.